Noname manuscript No.
(will be inserted by the editor)

Sampling from Dirichlet process mixture models with unknown concentration parameter: Mixing issues in large data implementations

David I. Hastie* · **Silvia Liverani*** · **Sylvia Richardson**

Received: date / Accepted: date

Abstract We consider the question of Markov chain Monte Carlo sampling from a general stick-breaking Dirichlet process mixture model, with concentration parameter α. This paper introduces a Gibbs sampling algorithm that combines the slice sampling approach of Walker (2007) and the retrospective sampling approach of Papaspiliopoulos and Roberts (2008). Our general algorithm is implemented as efficient open source C++ software, available as an R package, and is based on a blocking strategy similar to that suggested by Papaspiliopoulos (2008) and implemented by Yau et al (2011).

We discuss the difficulties of achieving good mixing in MCMC samplers of this nature and investigate sensitivity to initialisation. We additionally consider the challenges when an additional layer of hierarchy is added such that joint inference is to be made on α. We introduce a new label switching move and compute the marginal model posterior to help to surmount these difficulties. Our work is illustrated using a profile regression (Molitor et al, 2010) application, where we demonstrate good mixing behaviour for both synthetic and real examples.

Keywords Dirichlet process · mixture model · profile regression · Bayesian clustering

David I. Hastie
Imperial College London, UK

Silvia Liverani
Imperial College London, UK and
MRC Biostatistics Unit, Cambridge, UK

Sylvia Richardson
MRC Biostatistics Unit, Cambridge, UK
E-mail: sylvia.richardson@mrc-bsu.cam.ac.uk

* Joint first authors

1 Introduction

Fitting mixture distributions to model some observed data is a common inferential strategy within statistical modelling, used in applications ranging from density estimation to regression analysis. Often, the aim is not only to fit the mixture, but additionally to use the fit to guide future predictions. Approaching the task of mixture fitting from a parametric perspective, the task to accomplish is to cluster the observed data and (perhaps simultaneously) determine the cluster parameters for each mixture component. This task is significantly complicated by the need to determine the number of mixture components that should be fitted. An increasingly popular alternative approach to parametric modelling is to adopt a Bayesian non-parametric approach, fitting an infinite mixture, thereby avoiding determination of the number of clusters. The Dirichlet process (Ferguson, 1973) is a well studied stochastic process that is widely used in Bayesian non-parametric modelling, with particular applicability for mixture modelling. Draws from a Dirichlet process are themselves probability measures, with the property that all of their marginal distributions are Dirichlet distributions.

Our general algorithm, a DPMM (Dirichlet process mixture model) sampling algorthm, is implemented as efficient open source C++ code also available as an R package (Liverani et al, 2013). Our sampler is highly extensible, but currently permits sampling of Gaussian or categorical discrete mixtures, as well as more general problems, previously referred to collectively as "profile regression" (Molitor et al, 2010; Papathomas et al, 2011). The implementation also handles DPMM sampling with simultaneous variable selection, as discussed in Papathomas et al (2012).

Whilst previous methods have explored ways of sampling from the full DPMM, it appears that little discussion has been presented detailing the implicit difficulties of using a Gibbs (or Metropolis-within-Gibbs) sampling approach to update such a complex model space. In particular, for real (rather than synthetic) data applications of the DPMM, the state space can be highly multimodal, with well separated regions of high posterior probability co-existing, often corresponding to clusterings with different number of components. We demonstrate that such highly multimodal spaces present difficulties for the existing sampling methods to escape the local modes, with poor mixing resulting in inference that is influenced by sampler initialisation. In the most serious case, this can be interpreted as non-convergence of the MCMC sampler.

A secondary mixing issue relates to the mixing across the ordering of clusters in a particular clustering process. As we shall detail, such issues are particularly important when simultaneous inference is desired for the concentration parameter α. This mixing issue was highlighted by Papaspiliopoulos and Roberts (2008) who observed that the inclusion of label-switching moves can help to resolve the problem. We demonstrate that the moves that they propose offer only a partial solution to the problem, and we suggest an additional label switching move that appears to enhance the performance of the sampler.

In the following section, we present the details of the Dirichlet process mixture model, introducing our sampler for this model and the intricacies involved therein. This is followed by Section 3 where we present specific implementation details for various different mixture and profile regression models. Section 4 briefly discusses the post-processing methods that we use for the rich Bayesian output. Section 5 explores the issues of mixing for DPMM samplers, presenting a sensitivity analysis to sampler initialisation, followed by details of the computation of the marginal model posterior and label-switching functionality built in our sampler for resolving these issues. In Section 6 we present a demonstration of the applicability of our sampler to synthetic and real data examples, before conclusions are presented in Section 7.

2 Dirichlet process mixture models

A variety of ways have been used to show the existence of the Dirichlet Process, using a number of different formulations (Ferguson, 1973; Blackwell and MacQueen, 1973). In this paper we focus on Dirichlet process mixture models (DPMM), based upon the following simplified constructive definition of the Dirichlet process, due to Sethuraman (1994). If

$$\begin{aligned}
&P = \sum_{c=1}^{\infty} \psi_c \delta_{\Theta_c}, \\
&\Theta_c \sim P_{\Theta_0} \ \text{ for } c \in \mathbb{Z}^+, \\
&\psi_c = V_c \prod_{l<c} (1 - V_l) \ \text{ for } c \in \mathbb{Z}^+ \setminus \{1\}, \\
&\psi_1 = V_1, \quad \text{and} \\
&V_c \sim \text{Beta}(1, \alpha) \ \text{ for } c \in \mathbb{Z}^+,
\end{aligned} \tag{1}$$

where δ_x denotes the Dirac delta function concentrated at x, then $P \sim \text{DP}(\alpha, P_{\Theta_0})$. This formulation for $\boldsymbol{V}$ and $\boldsymbol{\psi}$ is known as a *stick-breaking* distribution. Importantly, the distribution P is discrete, because draws $\tilde{\Theta}_1, \tilde{\Theta}_2, \ldots$ from P can only take the values in the set $\{\Theta_c : c \in \mathbb{Z}^+\}$.

It is possible to extend the above formulation to more general stick-breaking formulations (Ishwaran and James, 2001; Kalli et al, 2011; Pitman and Yor, 1997).

2.1 Sampling from the Dirichlet process mixture model

More recently, two alternative innovative approaches to sampling the full DPMM have been proposed. The first, introduced by Walker (2007) and generalised by Kalli et al (2011), uses a novel slice sampling approach, resulting in full conditionals that may be explored by the use of a Gibbs sampler. The second distinct MCMC sampling approach was proposed in parallel by Papaspiliopoulos and Roberts (2008). The proposed sampler again uses a Gibbs sampling approach, but is based upon an idea termed *retrospective sampling*, allowing a dynamic approach to the determination of the number of components (and their parameters) that adapts as the sampler progresses. The cost of this approach is an ingeneous but complex Metropolis-within-Gibbs step, to determine cluster membership.

Despite the apparent differences between the two strategies, Papaspiliopoulos (2008) noted that the two algorithms can be effectively combined to yield an algorithm that improves either of the originals. The resulting sampler was implemented and presented by Yau et al (2011), and a similar version was presented by Dunson (2009). The current work presented in this paper is our interpretation of these ideas, implemented as efficient C++ code within the R package PReMiuM (Liverani et al, 2013) for general DPMM sampling.

For the Dirichlet process mixture model (DPMM), the (possibly multivariate) observed data $\boldsymbol{D} = (D_1, D_2, \ldots, D_n)$ follow an infinite mixture distribution, where component c of the mixture is a parametric density of the form $f_c(\cdot) = f(\cdot|\Theta_c, \Lambda)$ parametrised by some component specific parameter Θ_c and some global parameter Λ. Defining (latent) parameters $\tilde{\Theta}_1, \tilde{\Theta}_2, \ldots, \tilde{\Theta}_n$ as draws from a probability distribution P following a Dirichlet process $DP(\alpha, P_{\Theta_0})$ and again denoting the dirac delta function by δ, this system can be written,

$$D_i|\tilde{\Theta}_i, \Lambda \sim f(D_i|\tilde{\Theta}_i, \Lambda) \ \text{ for } i = 1, 2, \ldots, n, \tag{2}$$
$$\tilde{\Theta}_i \sim \sum_{c=1}^{\infty} \psi_c \delta_{\Theta_c} \ \text{ for } i = 1, 2, \ldots, n.$$

When making inference using mixture models (either finite or infinite) it is common practice to introduce a vector of latent allocation variables $\boldsymbol{Z}$. Such variables enable us to explicity characterise the clustering and additionally facilitate the design of MCMC samplers. Adopting this approach and writing $\boldsymbol{\psi} = (\psi_1, \psi_2, \ldots)$ and $\boldsymbol{\Theta} = (\Theta_1, \Theta_2, \ldots)$, we re-write Equation 2 as

$$D_i|\boldsymbol{Z}, \boldsymbol{\Theta}, \Lambda \sim f(D_i|\Theta_{Z_i}, \Lambda) \ \text{ for } i = 1, 2, \ldots, n,$$
$$\Theta_c \sim P_{\Theta_0} \ \text{ for } c \in \mathbb{Z}^+,$$
$$\mathbb{P}(Z_i = c|\boldsymbol{\psi}) = \psi_c \ \text{ for } c \in \mathbb{Z}^+, \ i = 1, 2, \ldots, n. \tag{3}$$

See Liverani et al (2013) for further details on our implementation of the Dirichlet process mixture model.

3 Example Models

The general sampler of the previous section is applicable for many specific models, depending on the choices of f and P_{Θ_0}. Our implementation allows for mixtures of Gaussian covariates, discrete covariates or a combination of the two, assuming independence between continuous and categorical data conditional on the cluster allocations (Liverani et al, 2013).

In particular, we are interested in using DPMM as an alternative to regression models, non-parametrically linking a response vector $\boldsymbol{Y}$ to covariate data $\boldsymbol{X}$ through cluster membership. This idea has been used by several authors including Dunson et al (2008), Bigelow and Dunson (2009), Molitor et al (2010), Papathomas et al (2011), and Molitor et al (2011). Our presentation is most similar to the latter three of these papers and we refer to it as *profile regression*. Formally, the data $\boldsymbol{D} = (\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{W})$ is now extended to contain response data Y_i and covariate data X_i for each individual i, where the contribution of the covariate data to the response may be cluster dependent. There is also the possibility to include additional fixed effects W_i for each individual, which are constrained to only have a global (i.e. non-cluster specific) effect on the response Y_i.

The data D_i is then jointly modelled as the product of a response model and a covariate model, to give the following likelihood:

$$p(D_i|Z_i, \boldsymbol{\Theta}, \Lambda) = f_Y(Y_i|\Theta_{Z_i}, \Lambda, W_i) f_X(X_i|\Theta_{Z_i}, \Lambda).$$

The likelihood f_Y depends upon the choice of response model. Our R package allows Bernoulli, Binomial, Poisson, Normal and categorical response, as well as Normal and discrete covariates.

4 Post processing

Given a sample of partitions from the posterior distribution of a Bayesian cluster model, where the sample is the output of an MCMC algorithm, it is often desirable to summarize the sample with a single representative clustering estimate. These include the maximum a posteriori (MAP) estimate and methods based on the posterior similarity matrix, a matrix containing the posterior probabilities that the observations i and j are in the same cluster. We find that methods based on the posterior similarity matrix are less suceptible to Monte Carlo error, especially when the optimal partition is obtained using clustering methods, such as partitioning around medoids, that take advantage of the whole MCMC output (Molitor et al, 2010).

Moreover, we also allow predicted values to be computed based on probabilistic allocations, or using a Rao-Blackwellised estimate of predictions, where the probabilities of allocations are used instead of performing draws.

5 Mixing of the MCMC algorithm

While the design and implementation of an MCMC sampler to sample the infinite DPMM has been the focus of previous research, little appears to be written about monitoring the performance of such a sampler in realistic applications, in particular with respect to investigating its convergence and mixing. Papaspiliopoulos and Roberts (2008) briefly discuss the importance of the inclusion of label switching to improve mixing across cluster orderings, and while we agree with the need for such moves (see Section 5.1), in our experience more fundamental mixing issues can affect the sampler for real-data applications. In this section we address these issues.

5.1 Ordering and the concentration parameter

One area that requires attention is the mixing of the algorithm over cluster orderings. In particular, whilst the likelihood of the DPMM is invariant to the order of cluster labels, the prior specification of the stick breaking construction is not. As detailed by Papaspiliopoulos and Roberts (2008), the definition of the ψ_c in terms of V_c, imposes the relation $\mathbb{E}[\psi_c] > \mathbb{E}[\psi_{c+1}]$ for all c. This weak identifiability, discussed in more detail by Porteous et al (2006), also manifests itself through the result $P(\psi_c > \psi_{c+1}) > 0.5$ for all c, a result that we prove in Appendix A.1.

The importance of whether the algorithm mixes sufficiently across orderings depends partially upon the object of inference. Specifically, since $P(\psi_c > \psi_{c+1})$ depends upon the prior distribution of α, if inference is to be simultaneously made about α (as is the scenario considered in this paper), it is very important that the algorithm exhibits good mixing with respect to α. If this was not the case, the posterior marginal distribution for α would not be adequately sampled, and since α is directly related to the number of non-empty clusters (see Antoniak,1974 for details), poor mixing across ordering may further inhibit accurate inference being made about the number of non-empty clusters. This situation would be further exaggerated for more general stick breaking constructions (of the sort mentioned in the introduction). While our sampler includes the possibility of setting a fixed value of α, more generally we wish to allow α to be estimated.

5.1.1 Label switching

To ensure adequate mixing across orderings, it is important to include label-switching moves, as observed by Papaspiliopoulos and Roberts (2008). Without such moves, the one-at-a-time updates of the allocations Z_i, mean that clusters rarely switch labels, and consequentially the ordering will be largely determined by the (perhaps random) initialisation of the sampler. For all choices of α, the posterior modal ordering will be the one where the cluster with the largest number of individuals has label 1, that with the second largest has label 2 and so on. However, α affects the relative weight of other (non-modal) orderings, and a properly mixing sampler must explore these orderings according to their weights.

We adopt the label-switching moves suggested by Papaspiliopoulos and Roberts (2008), and details can be found therein. However, in our experience, while these moves may experience high acceptance rates early on in the life of the sampler, once a "good" (in terms of high posterior support) ordering is achieved, the acceptance rates drop abruptly. This means that there is little further mixing in the ordering space. For the first of the moves that Papaspiliopoulos and Roberts (2008) propose, where the labels of two randomly selected clusters are exchanged, we observed acceptance rates below 10% for any sample of 500 sweeps. For the second of the moves, where the labels of two neighbouring clusters are swapped, along with the corresponding V_c, V_{c+1}, Figure 1 demonstrates the decrease in acceptance rate. This decrease can be explained by the observation (made by the original authors) that the second move type is always accepted if one of the clusters is empty, which can happen often in initial cluster orderings with low posterior support. However, our concern is that while these label-switching moves appear to encourage a move towards the modal ordering, once that ordering is attained, the sampler rarely seems to escape too far from this ordering.

Our solution is to introduce a third label switching move that we describe here. In brief, the idea is to simultaneously propose an update of the new cluster weights so they are something like their expected value conditional upon the new allocations. Specifically, defining $Z^\star = \max_{1\le i\le n} Z_i$ and $A = \{c \in \mathbb{Z}^+ : c \le Z^\star\}$ the move proceeds as follows: first choose a cluster c randomly from $A \setminus \{Z^\star\}$. Propose new allocations

$$Z'_i = \begin{cases} c+1 & i : Z_i = c \\ c & i : Z_i = c+1 \\ Z_i & \text{otherwise.} \end{cases} \tag{4}$$

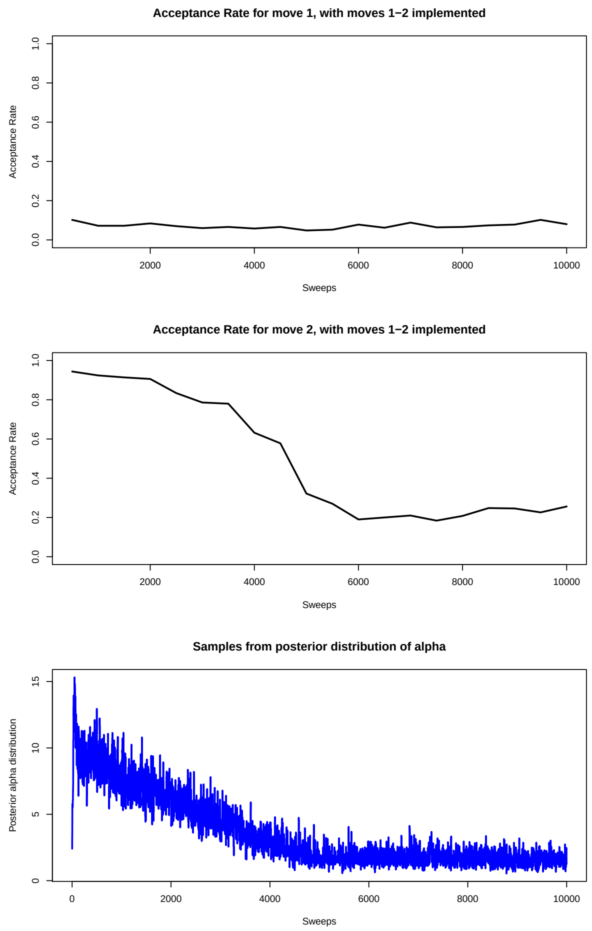

Fig. 1: Acceptance rate for intervals of 500 sweeps for the two label switching moves proposed by Papaspiliopoulos and Roberts (2008) and comparison with samples from the posterior distribution of α (bottom). Note that convergence appears to be achieved after 5,000 iterations for the example shown. If only the first of the two moves is implemented, alpha moves extremely slowly (more than 50,000 iterations are not enough, not shown) while if only the second of the two moves is implemented, for this example, 17,000 iterations are necessary for alpha to converge (not shown).

and switch parameters associated to these clusters such that

$$\Theta'_l = \begin{cases} \Theta_{c+1} & l = c \\ \Theta_c & l = c+1 \\ \Theta_l & \text{otherwise.} \end{cases} \tag{5}$$

Additionally, propose new weights ψ'_c and ψ'_{c+1} for components c and $c+1$ such that

$$\psi'_l = \begin{cases} \psi_{c+1} \frac{\psi^+}{\Psi'} \frac{\mathbb{E}[\psi_c|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z},\alpha]} & l = c \\ \psi_c \frac{\psi^+}{\Psi'} \frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z},\alpha]} & l = c+1 \text{ and} \\ \psi_l & \text{otherwise,} \end{cases} \tag{6}$$

where $\psi^+ = \psi_c + \psi_{c+1}$ and

$$\Psi' = \psi_{c+1} \frac{\mathbb{E}[\psi_c|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z},\alpha]} + \psi_c \frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z},\alpha]},$$

by setting

$$V'_l = \begin{cases} \frac{\psi'_c}{\prod_{l<c}(1-V_l)} & l = c \\ \frac{\psi'_{c+1}}{(1-V'_c)\prod_{l<c}(1-V_l)} & l = c+1 \\ V_l & \text{otherwise.} \end{cases} \tag{7}$$

All other variables are left unchanged. Assuming that there are n_c and n_{c+1} individuals in clusters c and $c+1$ respectively at the beginning of the update, the acceptance probability for this move is then given by $\min\{1, R\}$ where

$$R = \left(\frac{\psi^+}{\psi_c R_1 + \psi_{c+1} R_2} \right)^{n_c + n_{c+1}} R_1^{n_{c+1}} R_2^{n_c}, \quad \text{where} \tag{8}$$

$$R_1 = \frac{1 + \alpha + n_{c+1} + \sum_{l>c+1} n_l}{\alpha + n_{c+1} + \sum_{l>c+1} n_l}, \quad \text{and} \tag{9}$$

$$R_2 = \frac{\alpha + n_c + \sum_{l>c+1} n_l}{1 + \alpha + n_c + \sum_{l>c+1} n_l}. \tag{10}$$

More details can be found in Appendix A.2.

5.2 Sensitivity to initialisation and monitoring convergence

In practice, convergence is a key task of users of complex MCMC algorithms. For our sampler, with all the moves described in this paper fully implemented, simulated data with a strong underlying signal will convergence in a few iterations to the generating partition, even with substantially different initialisations. However, for real data, where the signal might not be very strong, different initialisations can result in substantially different partitions, indicating that convergence in the highly multi-modal model space remains a significant challenge.

Accepting that the challenge persists, it is clearly important that the user has diagnostic methods to assess whether convergence can be reasonably expected.. Due to the nature of the model space, many traditional techniques cannot be used in this context. For our hierarchical model, as described in Equations 1 and 3, there are no parameters that can be used to meaningfully demonstrate convergence of the algorithm. Specifically, the parameters of the fixed effects tend to converge really quickly, regardless of the underlying clustering, as they are not cluster specific and therefore are not a good indication of the overall convergence. On the other hand the cluster parameters, such as the θ_c's, cannot be tracked, as their number and interpretation changes from one iteration to the next (along with the additional complication that the labels of clusters may switch between iterations). While the concentration parameter α may appear to offer some information, using this approach can be deceiving, since a sampler that becomes stuck in a local mode in the clustering space will demonstrate a sample of α that appears to have converged.

Based upon our experience with real datasets, we suggest that to better assess convergence, it is important to compare multiple runs of the sampler started from significantly different initialisations, and to monitor the marginal model posterior in each run, a calculation that we detail in the following section.

Marginal Model Posterior We define the marginal model posterior as $p(\mathbf{Z}|\mathbf{D})$. This quantity represents the posterior distribution of the allocations given the data, having marginalised out all the other parameters. As such, it can be thought of a model posterior, as $\mathbf{Z}$ fully specifies the partition, which can be thought of as "the model" in our context. The marginal model posterior can be computed in closed form for discrete covariates but it requires some approximation when including the response model or Normal covariates. In practice, to reduce the complexity of the approximations required we have also conditioned on α. The choice of α to condition on is based on experiments on the dataset under study with α variable.

Computing the marginal model posterior for each run of the MCMC and comparing between runs has proven to be a very effective tool for our real examples, particularly to identify runs that were significantly different from others, perhaps due to convergence issues.

Figure 2 demonstrates that the strong signal in the simulation study in Liverani et al (2013) means that the sampler converges regardless of the initial number of clusters.

Initial Number of Clusters One consequence of our investigations using the marginal model posterior, was that despite the MCMC moves being very powerful, permitting significant moves across the model space, there is still considerable difficulties for the algorithm to split clusters and thereby escape local modes. This is due to the intrinsic characteristics

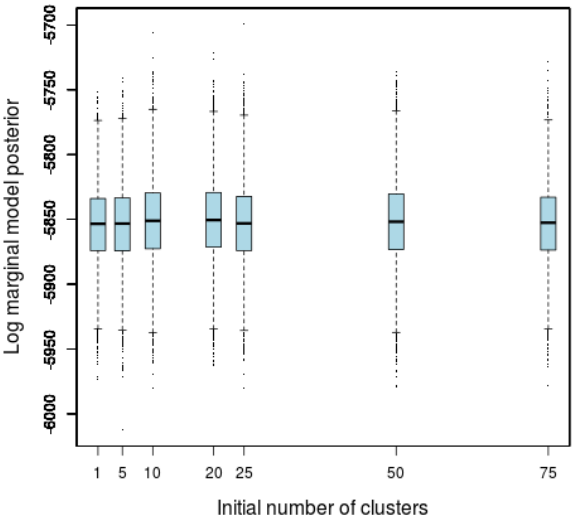

Fig. 2: Log marginal model posterior for the simulated dataset with different initial number of clusters.

of partition spaces and the extremely high number of possible ways to split a cluster, even if it only has a small number (for example, 50 or more) subjects in it.

Despite our best efforts to overcome these challenges, any Gibbs sampling based approach that updates (blocks of) parameters sequentially is likely to struggle to escape the local modes, due to the separation of these modes in model space. Furthermore, introducing more ambitious Metropolis-Hastings moves that attempt to update a larger number of parameters simultaneously is also a very difficult task due to the difficulty in designing moves to areas of the model space with similar posterior support.

Rather than subtly ignoring the problem, we suggest that used with caution our sampler still provides a useful inferential tool, but that the limitation must be realised and acknowledged. For example, because of the difficulty that the sampler has in increasing the number of clusters for problems with real data with weak signal, it is important to initialise the algorithm with a number of clusters which is greater than the anticipated number of clusters that the algorithm will convergence to. This necessarily involves an element of trial and error to determine what that number is, where multiple runs from different initialisations must be compared, for example using the marginal model posterior. This is demonstrated in Section 6.1.

Optimal Partitions vs Predictions Another important question when making inference using the Dirichlet process mixture model, is how best to interpret the output. Specifically, difficulties are faced in understanding or conveying the uncertainty of the partitioning. Many approaches make use of an “optimal” partition, which might be determined, for example, by using a similarity matrix based upon the output of the MCMC run. Accepting that this way of summarising the output may be important in many cases, our package allows such an approach, but in general we advise against its use where possible, as it is the equivalent of giving a point estimate without a posterior distribution or a credible interval, and therefore discards a significant amount of inferential information from the run. Moreover, due to the complexity and sheer size of the model space, the optimal partitions tend to differ between runs of the MCMC, and it is not an easy task to assess whether convergence has been achieved based on this approach alone.

In our experience, a better approach is to use posterior predictions, where posterior predictive distributions for quantities of interest can be derived from the whole MCMC run, taking the uncertainty over clustering into account. Depending on the quantity of interest, the posterior predictive distribution can often be relatively robust even across runs with noticeably different optimal partitions. While this may not help us to determine if the algorithm has sufficiently explored the model-space, if the purpose of the inference is to make predictions, this robustness can be reassuring.

6 Investigation of the algorithm’s properties in a large data application

In this section, we report the results of using our sampler on a real epidemiological dataset with 2,639 subjects. The figures and results presented here have been produced with our R package PReMiuM (Liverani et al, 2013). We do not report the results from a simulated study here as this is well documented in Liverani et al (2013) where it is shown that our algorithm performs well. In that case, while ensuring convergence is a complex problem, we have observed good stability in all our runs, with results from independent chains virtually identical. As expected, the analysis of real data is more challenging: it requires care in ensuring convergence, as the signal is not as strong as in a simulation study. However, these are challenges that might be encountered more widely by users wishing to apply the methods to real data, and by presenting an example and it allows us to highlight and discuss the issues that arise.

6.1 The data

Our dataset is a subset taken from an epidemiological case-control study, the analysis of which has provided the motivation of most of the work presented in this paper. In the illustrative example we have 2,639 subjects, and use 6 discrete covariates each with 5 categories. We also include 13

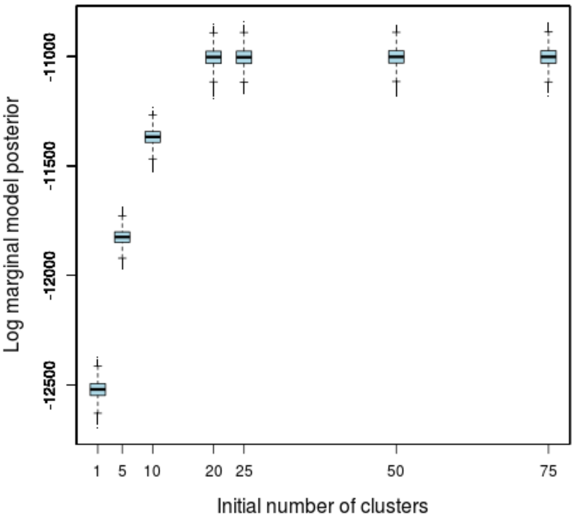

Fig. 3: Log marginal model posterior for the real epidemiological dataset with different initial number of clusters.

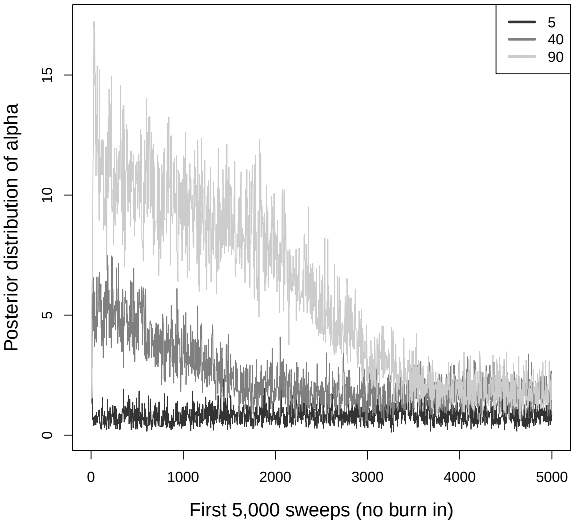

Fig. 4: Posterior distribution of α for different number of initial clusters: trace of the first 5,000 iterations for three different initialisation.

fixed effects. The low signal contained in the data poses issues with convergence of the MCMC, as we illustrate below.

Our results are based upon running the multiple chains each for 100,000 iterations after a burn-in sample of 50,000 iterations. In some cases, behaviour within this burn-in period is illustrated.

Marginal Model Posterior and Number of Clusters As discussed in Section 5 we run multiple MCMC runs, starting each with very different numbers of initial clusters. For this dataset, initialising the sampler with fewer than 20 clusters results in marginal model posterior distributions that are significantly different between runs. This is illustrated in Figure 3, where initialisations with small number of clusters result in much lower marginal model posterior values than can be achieved with a higher initial number of clusters. It is apparent that there is a clear cut-off at 20 clusters, where increasing the number of initial clusters further does not result in an increase in the marginal model posterior, suggesting that with 20 clusters or more the sampler is able to visit areas of the model space with the highest posterior support. This is further supported by additional comparisons of the predictions and parameters of the model runs (not reported), which provide additional of good mixing. For comparison, see the equivalent figure for the simulation study in Liverani et al (2013) shown in Figure 2.

Posterior Distribution of α Figure 4 shows the trace plots of α for a number of runs, each with a different initial number of clusters. As can be seen by considering the trace for the run with 5 initial clusters, if we were monitor to this trace in isolation, α appears to have converged. However, by comparing to the trace plots of initialisations with greater than 20 clusters, we see that it has "converged" to a different value than the value corresponding to models with higher posterior support (Figure 5). Monitoring the trace of α is not a good indication of convergence and inference on α relies on being able to properly explore the model space. This is particularly relevant for more complicated stick breaking constructions (as mentioned in the Section 1) where α may be replaced by a number of parameters.

As a final point on this matter, we observe that not unsurprisingly, the trace plots in Fig. 4 show that while it is advisable to start with a large number of initial clusters, starting with many more clusters than necessary can result in a larger number of iterations required for convergence.

Posterior distribution of the number of clusters The need to initialise the sampler with a sufficiently high number of clusters is supported by the behaviour in the burn-in period illustrated in Figure 6, for a run with 31 initial clusters. This plot shows the trace of the number of clusters for the first 500 iterations of the sampler. Figure 7 is the equivalent for the 500 iterations after the first 15,000. In the initial iterations, the space is explored by modifying and merging clusters, with the number of iterations changing frequently, in a general downward trend. On the other hand, once the MCMC has converged to the model space around a mode, the algorithm attempts to split clusters regularly, but the number of changes in the number of clusters are few, and increases in

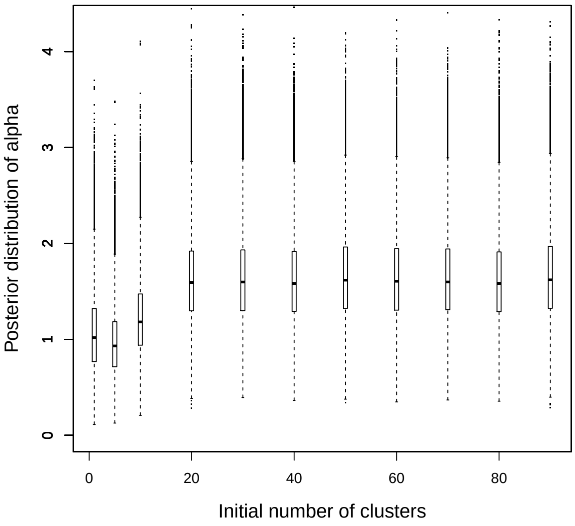

Fig. 5: Posterior distribution of α for different number of initial clusters: boxplots for the distribution after the burn-in.

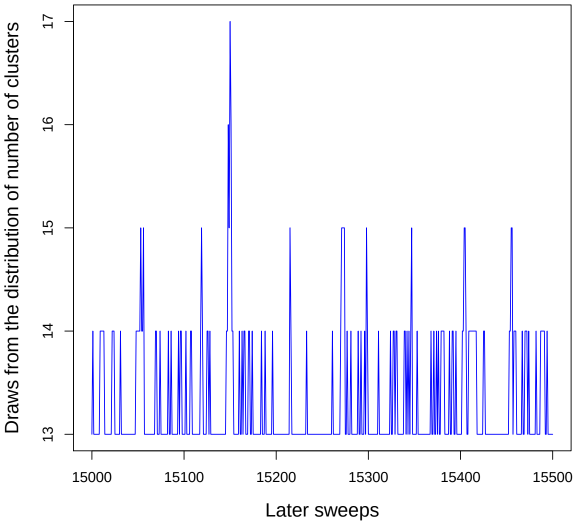

Fig. 7: The trace of the posterior of the number of clusters after 15,000 iterations of the MCMC.

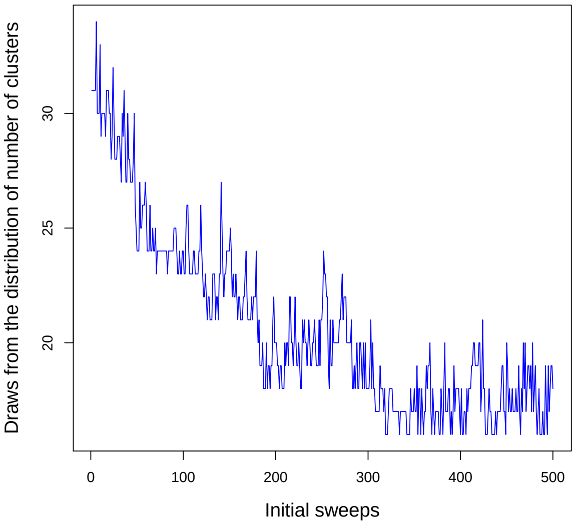

Fig. 6: The trace of the posterior of the number of clusters in the first 500 iterations of the MCMC.

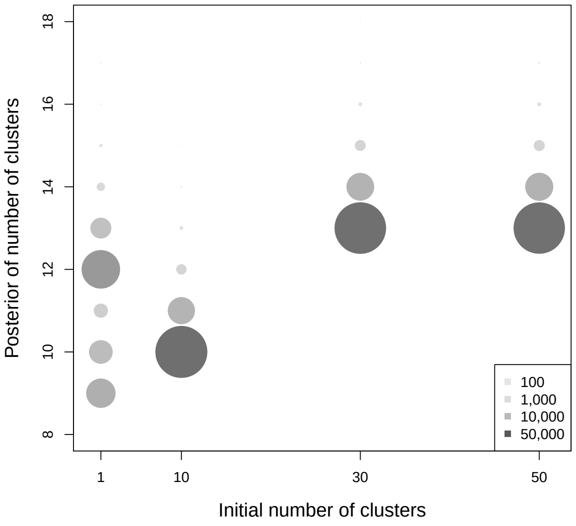

Fig. 8: The posterior distribution of the number of clusters for 50,000 sweeps after a burn-in of 50,000 iterations.

the number of clusters are almost immediately reversed in the following iteration.

The posterior distributions for the number of clusters is shown in Figure 8 for runs with different initial numbers of clusters. Here the size and shading of each circle represents the posterior frequency of each number of clusters. As can be seen from this figure, this provides further evidence that with 20 or more initial clusters the sampler will converge to a common area of posterior support, but with fewer than this the sampler will not visit this region of the model space, despite it having increased posterior support.

Label switching moves This example also demonstrates the need for the label switching move discussed in Section 5.1.1 to ensure good mixing. Comparing Figure 9 to Figure 1, we can see that the new label switching move suffers from no

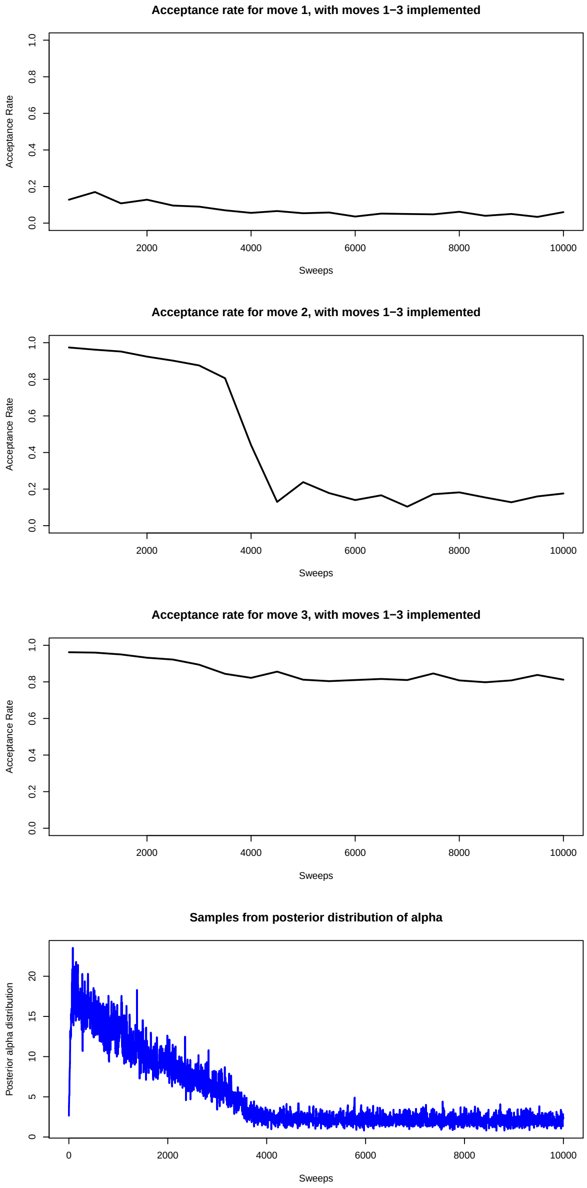

Fig. 9: Acceptance rates with the new label switching move.

drop off in acceptance at any point throughout the run and it converges in a smaller number of iterations. Figure 10 shows the acceptance rate for our new label switching move, when the other two switching label is not implemented. While the performance is worse than using all three moves, it is the most effective single label switching move (see Section 5.1.1). Thinning could be used to improve this, but the lack of convergence of α in Figure 11 when using only moves 1 and 2 demonstrates clearly the importance and efficacy of the label switching move.

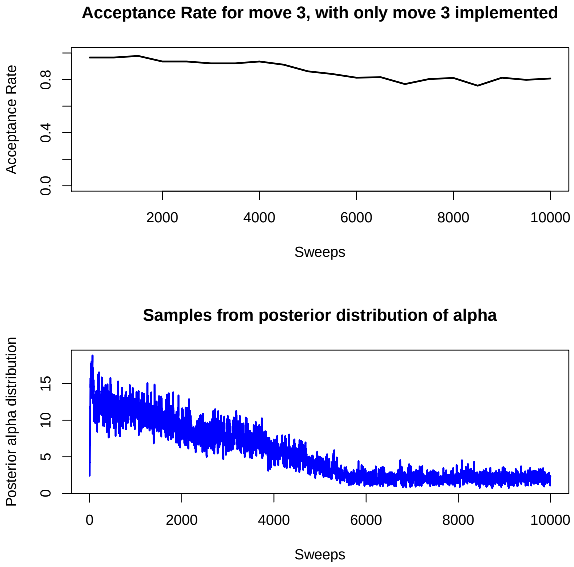

Fig. 10: Acceptance rates for the new label switching move.

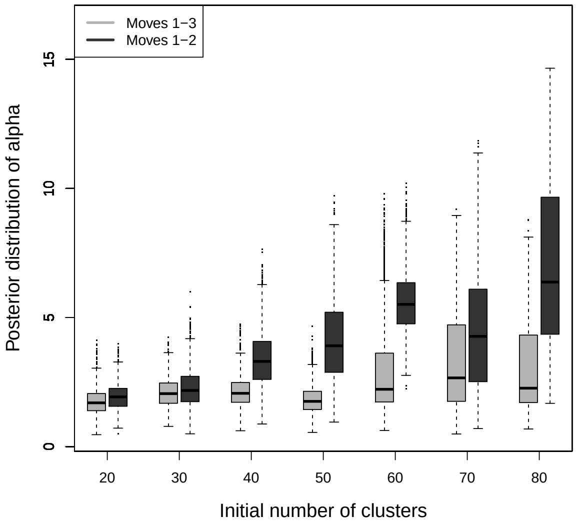

Fig. 11: Posterior of α with and without label switching moves.

7 Conclusions

Our implementation of an efficient C++ MCMC sampler for sampling Dirichlet process mixture models synthesizes many of the various techniques, such as parameter blocking, slice sampling, and label switching, introduced by several other authors and discussed within this paper, into a single piece of open source software, available as an R package.

However, sampling from the complex model space that is a feature of the Dirichlet process mixture model remains a challenging task. In previous work by other authors, considerable progress has been made evolving the samplers through innovative strategies and approaches. Nonetheless, discussion of many of the residual difficulties is avoided through demonstrating the methods only on simulated data, or for datasets with strong signal. In practice however, with real datasets, the user does not have the option of simply avoiding these issues.

In this paper we have attempted to highlight the difficulties that a user may face in practice. We have added our own new features to our sampler, such as new blocking strategies and additional label switching moves to build upon this previous research and further alleviate some of the challenges that are involved with such problems. We have also provided practical guidelines based on our experience, on how to make useful inference in the face of these limitations. Our work also demonstrates why it is important to not just ignore these issues, with specific focus on areas such as inference for stick breaking parameters that have not (to our knowledge) been previously discussed. Furthermore we have illustrated our discussions through the presentation of an example based upon an epidemiological dataset with a low signal to noise ratio.

As a consequence of specifying the challenges explicitly, we hope that our work will motivate further developments in this area to take additional steps to improve sampler efficiency. The challenge of designing MCMC moves that are able to escape local well-separated modes is considerable, but equally, so is the imagination and innovation of many practitioners developing new MCMC sampling methodologies.

Acknowledgements

Silvia Liverani acknowledges support from the Leverhulme Trust (ECF-2011-576). David I. Hastie acknowledges support from the INSERM grant (P27664). We are grateful for helpful discussions with Sara K. Wade and Lamiae Azizi.

A Appendices

A.1

We provide the following proposition concerning the relationship between the ordering and α.

Proposition 1 *Suppose that we have a model with posterior as given in Equation 1. Then $\mathbb{P}(\psi_c > \psi_{c+1}|\alpha)$ is a function of α, and furthermore $\mathbb{P}(\psi_c > \psi_{c+1}) > 0.5$.*

Proof *If $\psi_c > \psi_{c+1}$ then $V_c > V_{c+1}(1-V_c)$, which implies $V_{c+1} < V_c/(1-V_c)$. Thus*

$$\begin{aligned}\mathbb{P}(\ \psi_c > \psi_{c+1}|\alpha) &= \mathbb{P}(V_{c+1} < V_c/(1-V_c)|\alpha)\\ &= \int_0^{0.5}\int_0^{V_1/(1-V_1)} \alpha^2(1-V_1)^{\alpha-1}(1-V_2)^{\alpha-1}\mathrm{d}V_2\mathrm{d}V_1\\ &\qquad + \int_{0.5}^{1}\int_0^{1} \alpha^2(1-V_1)^{\alpha-1}(1-V_2)^{\alpha-1}\mathrm{d}V_2\mathrm{d}V_1\\ &= \int_0^{0.5}\left[\alpha(1-V_1)^{\alpha-1} - \alpha(1-V_1)^{\alpha-1}\left(\frac{1-2V_1}{1-V_1}\right)^{\alpha}\right]\mathrm{d}V_1\\ &\qquad + \int_{0.5}^{1}\alpha(1-V_1)^{\alpha-1}\mathrm{d}V_1\\ &= \int_0^{1}\alpha(1-V_1)^{\alpha-1}\mathrm{d}V_1\\ &\qquad - \int_0^{0.5}\alpha\frac{(1-2V_1)^{\alpha}}{1-V_1}\mathrm{d}V_1\\ &= 1 - \int_0^{0.5}\alpha\frac{(1-2V_1)^{\alpha}}{1-V_1}\mathrm{d}V_1.\end{aligned}$$

Now since, $(1-2V_1)^{\alpha}/(1-V_1) < (1-2V_1)^{\alpha-1}$

$$\alpha\int_0^{0.5}\frac{(1-2V_1)^{\alpha}}{1-V_1}\mathrm{d}V_1 < \alpha\int_0^{0.5}(1-2V_1)^{\alpha-1}\mathrm{d}V_1 = 0.5.$$

So $\mathbb{P}(\psi_c > \psi_{c+1}|\alpha) > 0.5$ *for all* α. *Finally,*

$$\begin{aligned}\mathbb{P}(\psi_c > \psi_{c+1}) &= \int \mathbb{P}(\psi_c > \psi_{c+1}|\alpha)p(\alpha)\mathrm{d}\alpha\\ &> \int 0.5p(\alpha)\mathrm{d}\alpha = 0.5.\end{aligned}$$

A.2

Proposition 2 *Consider the label switching move defined in Equations 4 to 7 in Section 5.1.1. Then:*

(i) $(\psi^+)' := \psi'_c + \psi'_{c+1} = \psi_c + \psi_{c+1} = \psi^+$;
(ii) $(1-V'_c)(1-V'_{c+1}) = (1-V_c)(1-V_{c+1})$;
(iii) The proposal mechanism is its own reverse;
(iv)

$$\frac{\mathbb{E}(\psi_c|\boldsymbol{Z}',\alpha)}{\mathbb{E}(\psi_{c+1}|\boldsymbol{Z},\alpha)} = \frac{1+\alpha+n_{c+1}+\sum_{l>c+1}n_l}{\alpha+n_{c+1}+\sum_{l>c+1}n_l} \quad \textit{and}$$

$$\frac{\mathbb{E}(\psi_{c+1}|\boldsymbol{Z}',\alpha)}{\mathbb{E}(\psi_c|\boldsymbol{Z},\alpha)} = \frac{\alpha+n_c+\sum_{l>c+1}n_l}{1+\alpha+n_c+\sum_{l>c+1}n_l}; \quad \textit{and}$$

(v) the acceptance probability for this move is given by $\min\{1,R\}$, *where the acceptance ratio R is given in Equation 8.*

Proof *(i) By definition*

$$\begin{aligned}(\psi^+)' &:= \psi'_c + \psi'_{c+1}\\ &= \frac{\psi^+}{\Psi'}\left(\psi_{c+1}\frac{\mathbb{E}[\psi_c|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z},\alpha]} + \psi_c\frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}',\alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z},\alpha]}\right)\\ &= \frac{\psi^+}{\Psi'}\Psi' = \psi^+;\end{aligned}$$

(ii) From (i),

$$\psi'_c + \psi_{c+1} = \psi_c + \psi_{c+1}$$

implies

$$\begin{aligned}\left[V'_c + V'_{c+1}(1-V'_c)\right] & \prod_{l<c}(1-V'_l)\\ &= [V_c + V_{c+1}(1-V_c)]\prod_{l<c}(1-V_l).\end{aligned}$$

By Equation 7, $V'_l = V_l$ for all $l < c$,

$$\Rightarrow \quad V'_c + V'_{c+1}(1 - V'_c) = V_c + V_{c+1}(1 - V_c)$$

$$\Rightarrow \quad (1 - V'_c)(1 - V'_{c+1}) = (1 - V_c)(1 - V_{c+1}).$$

The importance of this result is that it provides confirmation that our proposed ψ' in Equation 6 can be achieved with the V defined in Equation 7. In particular, with this choice of V', the only weights that are changed are those associated with components c and $c+1$, as desired.

(iii) Suppose that the Markov chain is currently in the proposed state defined in Equations 4 to 7 i.e. $(\boldsymbol{V}', \boldsymbol{\Theta}', \boldsymbol{Z}', \boldsymbol{U}, \alpha, \Lambda)$. We show that applying the proposal mechanism to this state, for component c and $c+1$, the proposed new state is the original state

$$(\boldsymbol{V}'', \boldsymbol{\Theta}'', \boldsymbol{Z}'', \boldsymbol{U}, \alpha, \Lambda) = (\boldsymbol{V}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{U}, \alpha, \Lambda.)$$

The parameters $\boldsymbol{U}$, α and Λ are unchanged by design of the proposal mechanism. Also, by design, the allocations $\boldsymbol{Z}$ and cluster parameters $\boldsymbol{\Theta}$ are simply swapped for the selected components, so trivially $\boldsymbol{Z}'' = \boldsymbol{Z}$ and $\boldsymbol{\Theta}'' = \boldsymbol{\Theta}$. Since V''_l is unchanged for $l \notin \{c, c+1\}$, it remains only to show $V''_c = V_c$ and $V''_{c+1} = V_{c+1}$, or equivalently $\psi''_c = \psi_c$ and $\psi''_{c+1} = \psi_{c+1}$. To confirm,

$$\begin{aligned}
\psi''_c &= \psi'_{c+1} \frac{(\psi^+)'}{\varPsi''} \frac{\mathbb{E}[\psi_c|\boldsymbol{Z}'']}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}', \alpha]} \\
&= \psi_c \frac{\psi^+}{\varPsi''} \frac{\psi^+}{\varPsi'} \frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}', \alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z}, \alpha]} \frac{\mathbb{E}[\psi_c|\boldsymbol{Z}'', \alpha]}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}', \alpha]} \\
&\quad (\textit{by (i) and Equation } 6) \qquad (11) \\
&= \psi_c \frac{(\psi^+)^2}{\varPsi''\varPsi'} \quad \textit{since } \boldsymbol{Z}'' = \boldsymbol{Z}. \qquad (12)
\end{aligned}$$

However,

$$\begin{aligned}
\varPsi'' &= \psi'_{c+1} \frac{\mathbb{E}[\psi_c|\boldsymbol{Z}'']}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}', \alpha]} + \psi'_c \frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}'', \alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z}', \alpha]} \\
&= \frac{\psi^+}{\varPsi'}(\psi_c + \psi_{c+1}) \\
&\quad (\textit{from Equation 6 and since } \boldsymbol{Z}'' = \boldsymbol{Z}) \\
&= \frac{(\psi^+)^2}{\varPsi'}.
\end{aligned}$$

Substituting this into Equation 12 we get $\psi''_c = \psi_c$. The result for ψ''_{c+1} can be shown by simply following identical logic.

(iv) From Equation 1, we have

$$\begin{aligned}
\mathbb{E}[\psi_c|\boldsymbol{Z}, \alpha] &= \mathbb{E}[V_c \prod_{l<c}(1 - V_l)|\boldsymbol{Z}, \alpha] \\
&= \mathbb{E}[V_c|\boldsymbol{Z}, \alpha] \prod_{l<c} \mathbb{E}[(1 - V_l)|\boldsymbol{Z}, \alpha] \\
&= \left(\frac{1 + n_c}{1 + \alpha + n_c + \sum_{l>c} n_l} \right) \qquad (13) \\
&\quad \times \prod_{l<c} \left(\frac{\alpha + \sum_{l'>l} n_{l'}}{1 + \alpha + n_l + \sum_{l'>l} n_{l'}} \right). \qquad (14)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}, \alpha] &= \left(\frac{1 + n_{c+1}}{1 + \alpha + n_{c+1} + \sum_{l>c+1} n_l} \right) \\
&\quad \times \left(\frac{\alpha + \sum_{l>c} n_l}{1 + \alpha + n_c + \sum_{l>c} n_l} \right) \qquad (15) \\
&\quad \times \prod_{l<c} \left(\frac{\alpha + \sum_{l'>l} n_{l'}}{1 + \alpha + n_l + \sum_{l'>l} n_{l'}} \right).
\end{aligned}$$

By definition of $\boldsymbol{Z}'$ in Equation 4, we have

$$n'_l = \begin{cases} n_{c+1} & l = c \\ n_c & l = c+1 \\ n_l & \textit{otherwise.} \end{cases} \qquad (16)$$

This means from Equations 13 and 15 we have

$$\begin{aligned}
\frac{\mathbb{E}[\psi_c|\boldsymbol{Z}', \alpha]}{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}, \alpha]} &= \left(\frac{1 + n'_c}{1 + \alpha + n'_c + n'_{c+1} + \sum_{l>c+1} n_l} \right) \\
&\quad \times \left(\frac{1 + \alpha + n_{c+1} + \sum_{l>c+1} n_l}{1 + n_{c+1}} \right) \qquad (17) \\
&\quad \times \left(\frac{1 + \alpha + n_c + n_{c+1} + \sum_{l>c+1} n_l}{\alpha + n_{c+1} + \sum_{l>c+1} n_l} \right)
\end{aligned}$$

Substituting Equation 16 into 17 and simplifying gives the desired results. The result for $\frac{\mathbb{E}[\psi_{c+1}|\boldsymbol{Z}', \alpha]}{\mathbb{E}[\psi_c|\boldsymbol{Z}, \alpha]}$ follows in the same fashion.

(v) By (iii) and the deterministic nature of the proposal mechanism, the only random feature of the proposal is the choice of component c. The probability of this choice is the same for the move and its reverse and so cancels. Therefore the only contribution to the acceptance ratio is the ratio of posteriors. By design, the likelihood is unchanged, and by (ii) the only change in posterior is down to the change in weights of components c and $c+1$. Therefore we have,

$$\begin{aligned}
R &= \frac{(\psi'_c)^{n'_c} (\psi'_{c+1})^{n'_{c+1}}}{\psi_c^{n_c} \psi_{c+1}^{n_{c+1}}} \qquad (18) \\
&= \left(\frac{\psi'_{c+1}}{\psi_c} \right)^{n_c} \left(\frac{\psi'_c}{\psi_{c+1}} \right)^{n_{c+1}} \quad \textit{by Equation 16.} \qquad (19)
\end{aligned}$$

Substituting in Equation 6 and the results in (iv), we obtain the desired acceptance ratio.

References

Antoniak CE (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. Annals of Statistics 2(6):1152–1174

Bigelow JL, Dunson DB (2009) Bayesian Semiparametric Joint Models for Functional Predictors. Journal of the American Statistical Association 104(485):26–36

Blackwell D, MacQueen JB (1973) Ferguson distributions via Polya Urn Schemes. Annals of Statistics 1(2):353–355

Dunson DB (2009) Nonparametric Bayes local partition models for random effects. Biometrika 96(2):249–262

Dunson DB, Herring AB, Siega-Riz AM (2008) Bayesian Inference on Changes in Response Densities Over Predictor Clusters. Journal of the American Statistical Association 103(484):1508–1517

Ferguson TS (1973) A Bayesian analysis of some nonparametric problems. Annals of Statistics 1(2):209–230

Ishwaran H, James LF (2001) Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association 96(453):161–173

Kalli M, Griffin JE, Walker SG (2011) Slice sampling mixture models. Statistics and Computing 21(1):93–105

Liverani S, Hastie DI, Richardson S (2013) PReMiuM: An R Package for Profile Regression Mixture Models using Dirichlet Processes, Preprint at arxiv.org, 1303.2836

Molitor J, Papathomas M, Jerrett M, Richardson S (2010) Bayesian profile regression with an application to the National Survey of Children's Health. Biostatistics 11(3):484–498

Molitor J, Su JG, Molitor NT, Rubio VG, Richardson S, Hastie D, Morello-Frosch R, Jerrett M (2011) Identifying vulnerable populations through an examination of the association between multipollutant profiles and poverty. Environmental Science & Technology 45(18):7754–7760

Papaspiliopoulos O (2008) A note on posterior sampling from Dirichlet mixture models. Tech. Rep. 8, CRISM Paper

Papaspiliopoulos O, Roberts GO (2008) Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. Biometrika 95(1):169–186

Papathomas M, Molitor J, Richardson S, Riboli E, Vineis P (2011) Examining the joint effect of multiple risk factors using exposure risk profiles: lung cancer in non-smokers. Environmental Health Perspectives 119:84–91

Papathomas M, Molitor J, Hoggart C, Hastie DI, Richardson S (2012) Exploring data from genetic association studies using Bayesian variable selection and the Dirichlet process : application to searching for gene × gene patterns. Genetic Epidemiology 6(36):663–74

Pitman J, Yor M (1997) The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. Annals of Probability 25(2):855–900

Porteous I, Ihler A, Smyth P, Welling M (2006) Gibbs sampling for (coupled) infinite mixture models in the stick breaking representation. In: Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06), AUAI Press, Arlington, VA

Sethuraman J (1994) A constructive definition of Dirichlet priors. Statistica Sinica 4:639–650

Walker SG (2007) Sampling the Dirichlet mixture model with slices. Communications in Statistics - Simulation and Computation 36:45–54

Yau C, Papaspiliopoulos O, Roberts GO, Holmes C (2011) Bayesian non-parametric hidden Markov models with applications in genomics. Journal of the Royal Statistical Society, Series B (Statistical Methodology) 73:37–57